

# OSPF – Open Shortest Path First

---



- Fundamentos
- Áreas
- Comunicação entre routers OSPF
- Vizinhos, Adjacências e *Designated Routers*
- Bases de dados de LSAs: Sincronização e *Flooding*
- Rotas intra-área
- Rotas inter-área
- Rotas inter-AS
- Exemplos: rotas intra-área
- Exemplos: rotas inter-área

# Estrutura de dados da área (1)



- Contém toda a informação utilizada para correr o algoritmo básico de encaminhamento do OSPF.
  - Cada área é composta por uma ou mais sub-redes em que cada *router* mantém a sua própria *link-state database*.
- Numa área, as *link-state database* existentes nos vários *routers*, são todas iguais entre si.
- Uma sub-rede pertence apenas a uma área.
- Uma interface dum *router* liga apenas a uma única área.
- Cada adjacência de um *router* é efectuada dentro de uma única área.
- O *backbone* OSPF (área 0) é uma área especial responsável por disseminar informação de encaminhamento entre as várias áreas.

# Estrutura de dados da área (2)



- *Area ID*
- Lista da gama de endereços da área
  - Esta lista é constituída por pares [endereço, máscara] e por uma indicação de estado *Advertise* ou *DoNotAdvertise*
- Interfaces associadas ao *router* (*associated router interfaces*)
  - Interfaces do *router* que ligam à área. Uma interface pertence a uma e apenas a uma área. No caso do *backbone* a área inclui os *virtual links*. Um *virtual link* é identificado pelo *router ID* do ponto onde termina e o seu custo é o custo do caminho mais curto (intra-area) através da área de trânsito atravessada entre os dois *routers*.
- Lista de *router*-LSA
- Lista de *network*-LSA
- Lista de *summary*-LSA

# Estrutura de dados da área (3)



- *Shortest-path tree*
  - Árvore dos caminhos mais curtos com o próprio *router* como raiz. Calculada a partir da colecção de *router*-LSA e *network*-LSA pelo algoritmo de Dijkstra.
- *TransitCapability*
  - *True* se puder ser atravessada por *virtual links*
- *ExternalRoutingCapability*
  - Se é possível propagar os *AS-external*-LSA para/através da área
- *StubDefaultCost*
  - Se a área foi configurada como uma área de *stub* e o *router* for um ABR então este parâmetro indica o custo do *summary*-LSA por omissão que o *router* deve anunciar nessa área.



- É uma área que se encontra no ramo final da árvore da rede do AS
- Tem como principal característica não receber LSA externos
- Tem como vantagem a redução da base de dados de LSA
- Características
  - ABR não injecta LSA External (Type 5) na área
  - ABR injecta (automaticamente) uma rota por omissão na área como sendo LSA Type 3
  - Não pode conter nenhum ASBR
  - Não pode ser atravessada por *Virtual Links*

# Totally Stubby Areas



- Trata-se de uma área que não recebe AS-External-LSA (tipo 5)
- Trata-se de uma área que não recebe Summary-LSA (tipo 3 e 4)
- Características
  - ABR não injecta AS-external-LSA (Type 5) na área
  - ABR não injecta ASBR Summary-LSA (Type 4) na área
  - ABR não injecta Summary-LSA (Type 3) na área
  - ABR injecta (automaticamente) uma rota por omissão na área
  - Não pode conter ASBRs
  - Não pode ser atravessada por *Virtual Links*
  - Também chamadas “Stub no-Summaries Areas”

# Not So Stubby Areas (NSSA )



- É uma área que pode receber AS-External-LSA tipo 7 mas não do tipo 5
- Serve para discriminar entre rotas externas que devem ser anunciadas e outras que são igualmente externas mas não devem ser anunciadas
- Características
  - AS-external-LSA de outras áreas não são propagados para a área
  - É automaticamente propagada uma rota por omissão para a área (LSA Type 3)
  - Pode conter ASBR
  - Propaga AS-external-LSA com rotas dos seus ASBR para as outras áreas
  - Não pode ser atravessada por Virtual Links



- Liga uma NSSA X ao *Backbone* (Área 0)
- Fluxo de LSA: **NSSA X » Backbone**
  - Gera LSA tipo 3 com informação dos LSA tipo 1 e 2 locais
  - Gera LSA tipo 4 por cada LSA tipo 7 dum ASBR local
  - Converte LSA tipo 7 em LSA tipo 5
- Fluxo de LSA: **Backbone » NSSA X**
  - Filtra LSA tipo 4 e 5
  - Gera LSA tipo 3 com a rota por omissão (0.0.0.0/0) para representar a informação externa gerada pelos ASBR (LSA tipo 4 e 5)
  - Gera LSA tipo 3 com informação dos LSA tipo 1 e 2 locais
    - Podem ser filtrados configurando “no-summaries”
  - Propaga os LSA tipo 3 gerados por ABR de outras áreas
    - Podem ser filtrados configurando “no-summaries”



# OSPF – Open Shortest Path First



- Fundamentos
- Áreas
- Comunicação entre routers OSPF
- Vizinhos, Adjacências e *Designated Routers*
- Bases de dados de LSAs: Sincronização e Flooding
- Rotas intra-área
- Rotas inter-área
- Rotas inter-AS
- Exemplos: rotas intra-área
- Exemplos: rotas inter-área

# Comunicação entre *routers* OSPF



- As mensagens OSPF são enviadas em datagramas IP
  - Utilizam no datagrama IP o código de protocolo 89
  - Sem especificar nenhum Type of Service TOS = 0
  - Preference = *Internetwork Control* (se possível)
- As mensagens OSPF são enviadas directamente de um *router* para outro(s) no mesmo segmento (sub-rede)
- Retransmissões de pacotes *Link State Update* são sempre enviadas directamente para o vizinho
  - Em redes *multiple access* as retransmissões são enviadas para o endereço IP do vizinho

Em OSPF utiliza-se ***multicast*** sempre que possível:

**224.0.0.5** ***AllSPFRouters (All Shortest Path First Routers)***, e

**224.0.0.6** ***AllDRouters (All Designated Routers)***

# Comunicação entre *routers* OSPF



- Nas redes com suporte de *multicast* o modo de transmissão preferencial é o *multicast*. Nestas redes os pacotes Hello são enviados para o endereço *multicast* AllSPFRouters, tal como fazem o DR e o BDR quando enviam Link State Updates e e *Link State Acknowledgment*.
  - Todos os outros *routers* enviam os seus *Link State Updates* e *Link State Acknowledgments* para o endereço AllDRRouters.
- Noutros tipos de redes incluindo *virtual links* a maioria das mensagens OSPF são enviadas por *unicast*, sendo o endereço IP destino o dos seus *routers* adjacentes
- Em ligações ponto-a-ponto o endereço destino é sempre AllSPFRouters

# Mensagens OSPF - *Header*



- **Type:** (1) *Hello*; (2) *Database Description*; (3) *Link Status Request*; (4) *Link Status Update*; (5) *Link Status Acknowledgement*
- **Router ID:** Identidade do *router* de origem da mensagem
- **Area ID:** Área de origem da mensagem
- **Checksum:** Complemento para 1 de todo o pacote, excluindo o campo *AuthenticationType*
- **AuType e Authentication Data:** AuType indica o tipo de autenticação em uso. O campo de autenticação de 64 bits é para uso do processo de autenticação.

Version	Type	Packet Length
Router ID		
Area ID		
Checksum		Authentication Type
Authentication Data		
Authentication Data		

# Autenticação das mensagens OSPF

---



- Os pacotes OSPF podem utilizar autenticação.
- Interfaces distintas, ligadas a redes distintas, podem utilizar autenticações distintas
- Troca de informação autenticada entre *routers* de modo a evitar ataques
- Tipos de autenticação
  - None – Sem autenticação
  - Simple - Chave em claro
  - MD5 – *Hash* da chave
- Podem existir esquemas de autenticação diferentes consoante a área

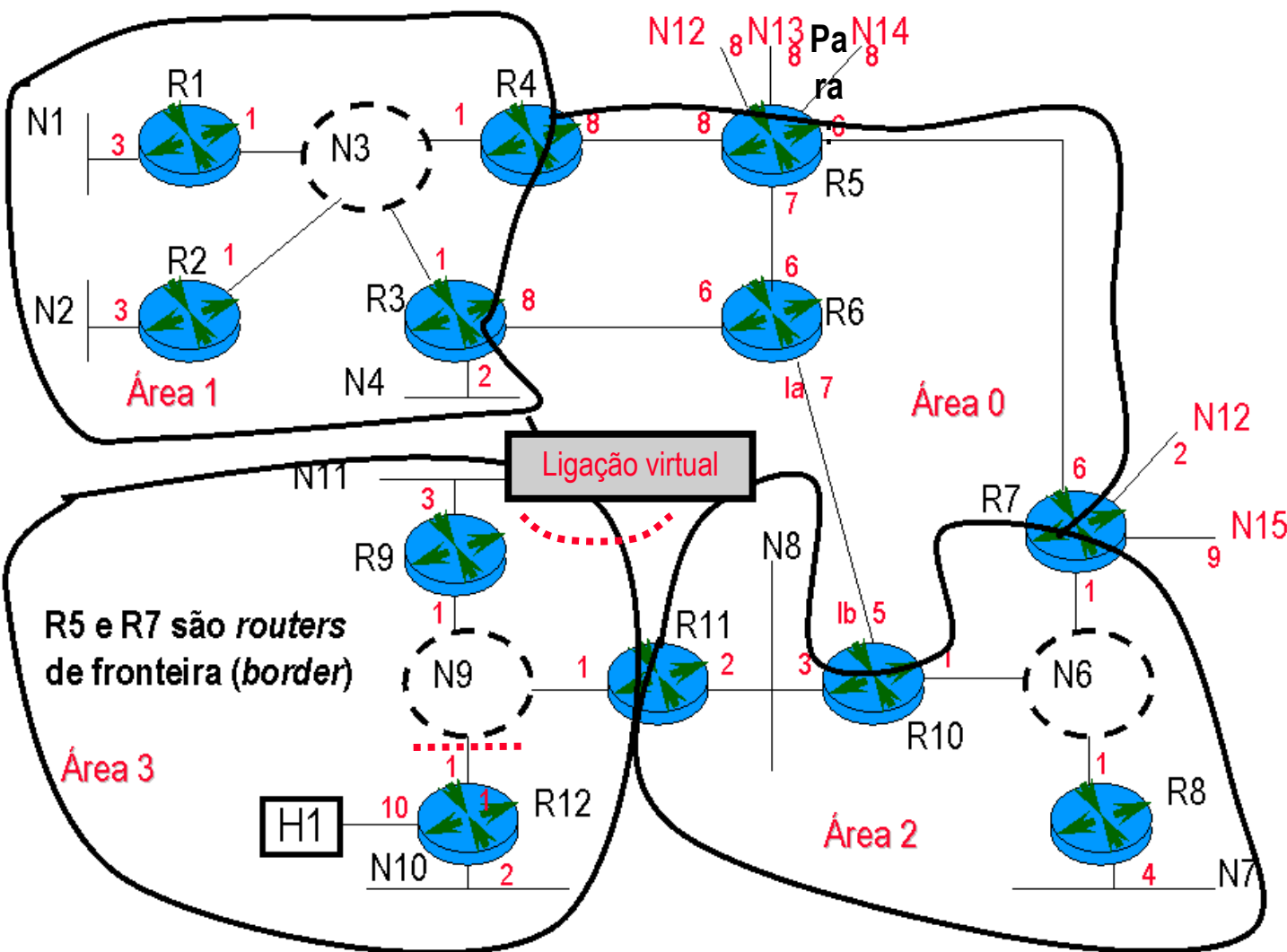
# Recepção de mensagens

---



- Se o endereço destino do pacote IP for *AllDRouters* este só deve ser aceite se o estado do *router* for DR ou *BDR*
- A mensagem deve ser autenticada
- Se a mensagem for do tipo Hello deve ser processada pelo protocolo Hello
- Todas as outras mensagens são trocadas apenas entre *routers* adjacentes
- Isto implica que deve ser proveniente de um dos vizinhos activos. Se não for é descartada
- Em redes com *broadcast*, ponto-a-multiponto e NBMA quem envia é indicado no endereço IP de origem do datagrama
- Em redes ponto-a-ponto ou ligações virtuais a origem é indicada pelo campo *Router ID* do *header* da mensagem OSPF

# Recepção de mensagens



Quando é recebida uma mensagem ela é marcada com a identificação da interface por onde foi recebida. Para *routers* que possuam *links* virtuais pode não ser óbvio de onde proveio a mensagem. Por exemplo, considere-se que R11 recebe uma mensagem pela sua interface com N8, ele pode querer associar a mensagem com a área 2 ou com a ligação virtual com R10.

Para o que se segue assume-se que a mensagem não é associada à ligação virtual.

# Recepção de mensagens

---



- Para a mensagem ser aceite ao nível IP deve:
  - O *checksum* IP deve estar correcto
  - O endereço IP deve ser o da interface que está a receber ou um dos endereços *multicast* AllSPFRouters ou AllDRouters
  - O protocolo especificado no pacote IP deve ser OSPF
  - Os pacotes gerados localmente não devem ser passados ao OSPF.



# Recepção de mensagens

---



- A seguir o *header* OSPF deve ser verificado:
  - A versão deve ser 2
  - O campo Area ID deve:
    - Ser igual ao Area ID da interface que recebe. Em redes que não ponto-a-ponto, após aplicar a máscara deve ser verificado se o endereço de rede é igual. Neste caso a mensagem atravessou um simples troço de rede.
    - Indicar o *backbone*. Neste caso foi enviado através duma ligação virtual. O *router* que recebe deve ser um ABR e o endereço de origem da mensagem deve ser o da outra ponta da ligação virtual.



- São necessários dois tipos de *timers*:
  - **single shot timers** - disparam uma vez e provocam que um evento do protocolo seja processado
  - **interval timers** - disparam a intervalos regulares. São utilizados para enviar pacotes a intervalos regulares
    - Exemplo: Envio de pacotes Hello
- A granularidade dos *timers* é de um segundo
- No envio de pacotes deve-se evitar a sincronização entre *routers* para evitar sobrecargas pontuais da rede. Para evitar este problema cada *router* introduz pequenos atrasos aleatórios.

# OSPF – Open Shortest Path First



- Fundamentos
- Áreas
- Comunicação entre routers OSPF
- Vizinhos, Adjacências e *Designated Routers*
- Bases de dados de LSAs: Sincronização e *Flooding*
- Rotas intra-área
- Rotas inter-área
- Rotas inter-AS
- Exemplos: rotas intra-área
- Exemplos: rotas inter-área

# Comunicação entre *routers* vizinhos



- As mensagens de Hello são responsáveis por estabelecer e manter as relações entre vizinhos garantindo que existe comunicação bidireccional.
- As mensagens de Hello contêm, entre outros, os seguintes campos:
  - Prioridade do *router* que envia
  - Intervalo entre mensagens de Hello
  - O RouterDeadInterval – intervalo máximo entre mensagens de Hello
  - Máscara da rede por onde é enviado (pode ser 0.0.0.0 em redes ponto-a-ponto não numeradas ou em ligações virtuais)
  - O campo de opções define as capacidades adicionais do *router*. O bit E indica se a área é capaz de processar os AS-external-LSA, ou seja, não é uma área *stub*.
- Em redes *broadcast* e ponto-a-ponto: *multicast* – ALLSPFRouters
- Em ligações virtuais: *unicast* – endereço da outra ponta da ligação
- Em rede ponto-a-multiponto: *unicast* - mensagens Hello independentes em cada ligação

# Protocolo Hello (ligações ponto-a-ponto)



- Objectivo
  - Responsável por estabelecer e manter as relações entre vizinhos garantindo que existe comunicação bidireccional.
- Funcionamento
  - O *router* envia mensagens Hello (periódicas) para a rede
    - Usa *Multicast* IP (224.0.0.5) - AllSPFRouters
    - Indica na mensagem enviada os endereços dos vizinhos que já conhece.
  - Recebe mensagens Hello dos *routers* vizinhos
    - Se o seu endereço vier listado na mensagem recebida, existe comunicação bidireccional com o *router* em causa. Isto é, está estabelecida uma relação de vizinhança.

# Protocolo Hello (redes BMA)



- Objectivo
  - Estabelecer e manter as relações entre vizinhos garantindo que existe comunicação bidireccional.
  - Eleger o *Designated Router* (DR) e o *Backup* DR de cada segmento
  - O DR determina quais os *routers* que se devem tornar adjacentes
  - Funcionar como “*keepalive*”
- Funcionamento
  - *Router* envia mensagens Hello (periódicas) para a rede
    - Usa *multicast* IP *AllSPFRouters* e *multicast* MAC
    - Indica na mensagem que envia os endereços dos vizinhos que já conhece
    - Indica a sua perspectiva sobre quem é o DR e o BDR
  - Recebe mensagens Hello dos *routers* vizinhos
    - Se o seu endereço vier indicado na mensagem recebida, existe comunicação bidireccional com o *router* que enviou a mensagem pelo que está estabelecida a relação de vizinhança.
    - O *router* que tiver maior prioridade ou o maior *router* ID, se as prioridades forem iguais, será o DR.

# Protocolo Hello (redes NBMA )



- Objectivo
  - Responsável por estabelecer e manter as relações entre vizinhos garantindo que existe comunicação bidireccional
  - Eleger o Designated Router (DR) e o Backup DR (BDR) da rede
  - O DR determina quais os *routers* que se devem tornar adjacentes
- Funcionamento
  - *Router* que quer ser DR envia mensagens Hello para uma lista de *routers* seus vizinhos que podem vir a ser DR
    - Indica a sua perspectiva sobre quem é o DR e o BDR
  - Recebe mensagens Hello de outros *routers* que podem ser DR
    - Se outro *router* tiver maior prioridade ou prioridade igual e router ID maior passa a ser DR
  - O DR envia e recebe de todos os outros *routers* mensagens Hello
    - Indica os endereços dos vizinhos que já conhece

# Mensagem Hello



Version	Type = 1	Packet Length	
Router ID			
Area ID			
Checksum		Authentication Type	
Authentication Data			
Authentication Data			
Network Mask			
Hello Interval		Options	Router Prio.
Router Dead Interval			
Designated Router ID			
Backup Designated Router ID			
Neighbout			



# Mensagem *Hello*



Formato do campo *Options*:

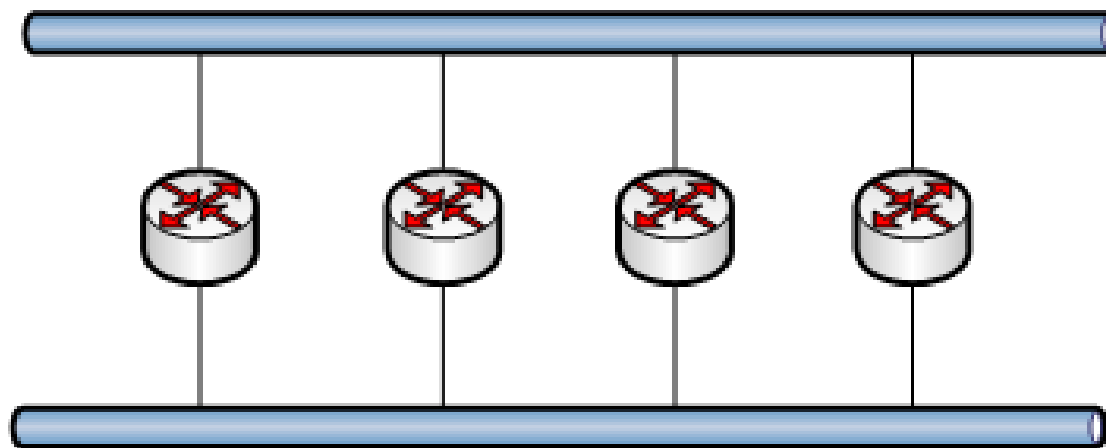


- **E-bit** – descreve a forma como os AS-external-LSA são enviados
- **MC-bit** – descreve a forma como os pacotes IP *multicast* são enviados [RFC 1584]
- **N/P-bit** – descreve como são tratados os Type-7 LSA (NSSA) [RFC 1587]
- **EA-bit** – descreve a disponibilidade do *router* para receber e enviar AS-*external-LSA*
- **DC-bit** – descreve a forma como o *router* trata o pedido de circuitos [RFC 1793]

# Routers vizinhos e adjacentes



- $N$  routers na mesma rede (*broadcast* ou *non-broadcast*)
  - $N(N-1)$  LSA serão necessários para transmitir informação sobre a mesma rede.



- A forma de minimizar a sobrecarga na rede devido aos LSA é eleger um DR entre os *routers* vizinhos, definindo as adjacências apenas entre estes e o DR e o BDR e não entre todos os vizinhos



- As adjacências são estabelecidas com algum subconjunto dos *routers* vizinhos
- Routers ligados a redes ponto-a-ponto, redes ponto-a-multiponto e ligações virtuais tornam-se sempre adjacentes
- Em redes BMA e NBMA todos os *routers* se tornam adjacentes do DR e BDR.
- **Uma adjacência deve ser estabelecida com um vizinho sempre que:**
  - O tipo da rede for ponto-a-ponto, ponto-a-multiponto ou uma ligação virtual
  - O *router* for DR ou BDR
  - O *router* vizinho for DR ou BDR

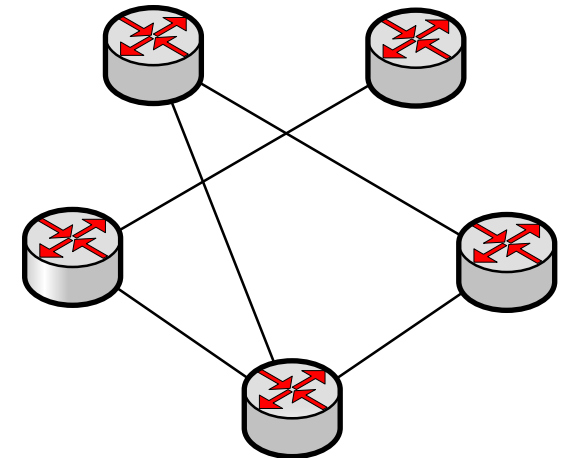
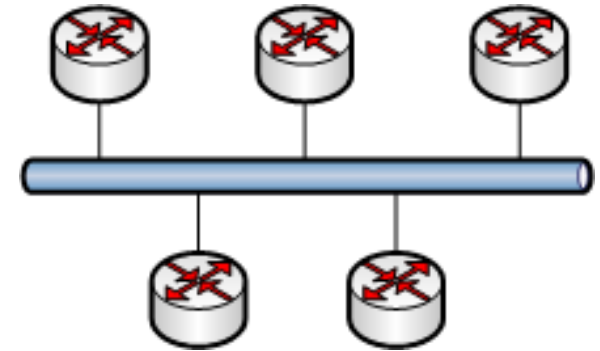


- *Routers* adjacentes, para além da troca das mensagens de Hello, **participam na troca da base de dados** (“mapa”) da área (sincronização da *link state database*).
- Para minimizarem o tráfego elegem um *designated router* (DR) e um *backup designated router* (BDR) por cada segmento multi-acesso da área.
- O DR funciona como *relay* da informação para os seus *routers* adjacentes.

# Tipos de redes



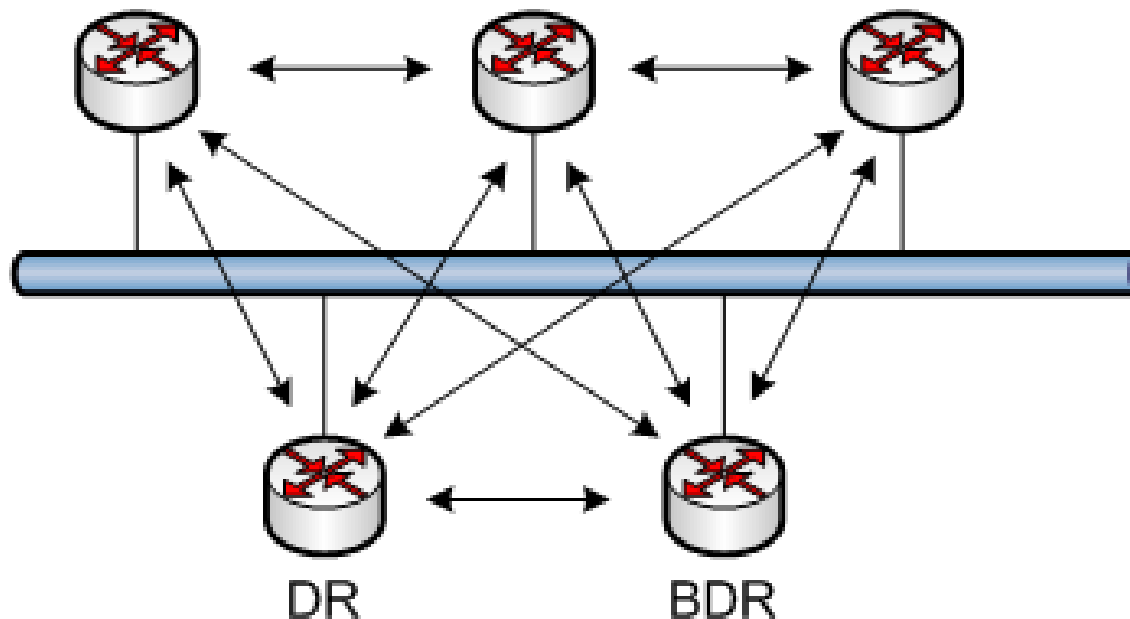
- **Ponto-a-ponto (point-to-point)**
  - Ex.: Ligação série
  - Não tem DR ou BDR
- **BMA (Broadcast Multiple Access)**
  - Ex.: Ethernet, Token Ring, FDDI
  - Utiliza *multicast* e são eleitos DR e BDR
    - 224.0.0.5 AllSPFRouters
    - 224.0.0.6 AllDRouters
- **NBMA (Non-Broadcast Multiple Access)**
  - Ex.: FR, ATM, X.25
  - São definidos DR e BDR
- **Ponto-a-multiponto (point-to-multipoint)**
  - NBMA tratada como múltiplos ponto-a-ponto
  - Não tem eleição de DR e BDR



# Designated Router



- Em redes **NBMA** e **Broadcast Multi-Access** (BMA) são escolhidos um **Designated Router** (DR) e um **Backup Designated Router** (BDR)



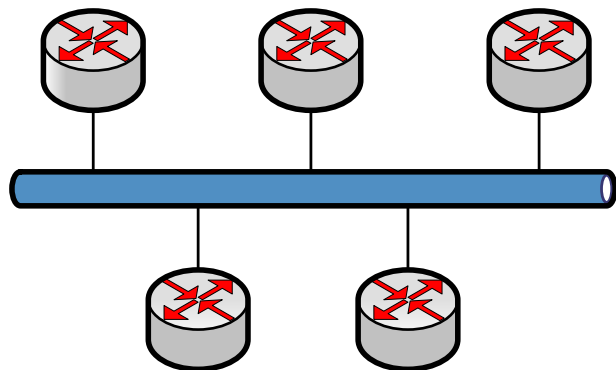
- A escolha do BDR e do DR é efectuada pelo protocolo Hello
  - A prioridade e o ID do *router* são usados na eleição
  - Router priority* igual a 0 significa que o *router* não pode ser DR nem BDR

# Funções do DR e do BDR

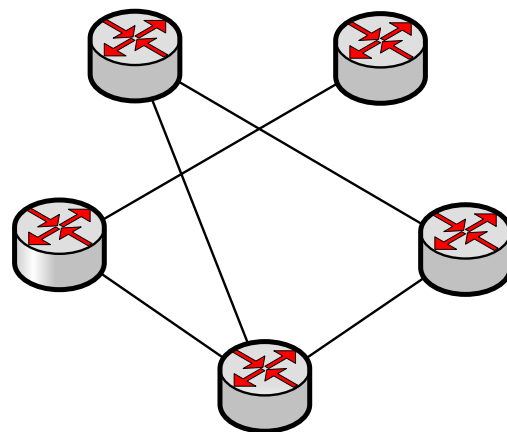


- DR e BDR: Criam uma adjacência com cada *router* do respectivo segmento
  - Sincronizam as bases de dados de *link state* com todos os *routers* adjacentes
- DR: Gera os *network LSA* em nome dos *routers* da rede
  - Usando como ID o endereço IP do DR na rede
  - Envia uma lista de todos os *routers* do segmento

**Redes Broadcast**



**Redes NBMA**



# Algoritmo de eleição do *Designated Router* (DR)



- A primeira vez que o *router* corre o algoritmo, o DR e o BDR são inicializados a 0.0.0.0 nas mensagens de Hello, isto indica a não existência de DR ou BDR.
- A lista de vizinhos do *router* X que se encontram num estado igual ou superior a 2-way é examinado. São excluídos todos os *routers* não elegíveis para DR (exemplo: prioridade = 0).
- Tendo em consideração a lista resultante, onde se inclui o *router* X, os próximos passos são os seguintes:
  1. É anotado quem são os actuais BDR e DR.
  2. Calcula-se o novo BDR de entre os que declaram não serem DR. Se houver *routers* a declararem-se BDR (nas mensagens de Hello) o que tiver maior prioridade versus Router ID será eleito BDR. Se não houver nenhum a declarar-se BDR a eleição será efectuada entre todos os elegíveis que não se tenham declarado DR. Será eleito o que possuir maior Prioridade versus Router ID.



# Algoritmo de eleição do *Designated Router* (DR) (cont.)

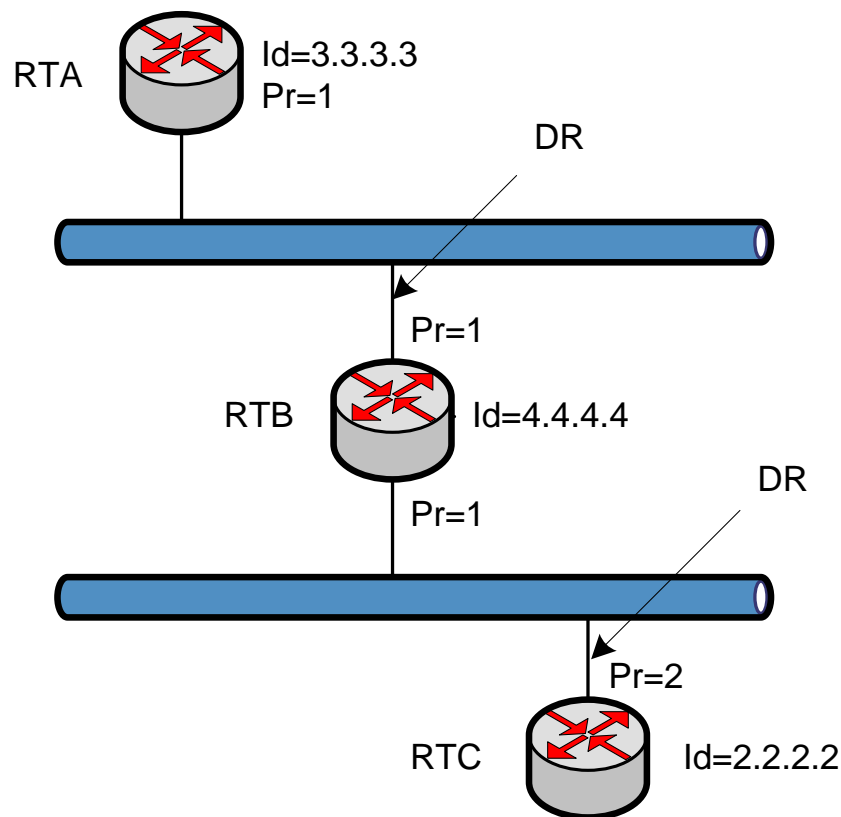


3. O novo DR será então eleito então da seguinte forma: De entre os *routers* que se declararam DR é eleito o que tiver maior prioridade versus *router* ID. Se não houver nenhum que se tenha declarado DR o novo DR será igual ao BDR eleito.
4. Se o *router* X passar a ser agora o novo DR ou BDR ou deixar de ser o BDR ou DR repetir os passos 2 e 3 e prosseguir para 5. Por exemplo, se o *router* X for agora o DR quando o passo 2 for repetido ele não entrará para a eleição do BDR. Entre outras coisas evita-se assim que um *router* seja BDR e DR em simultâneo.
5. Como resultado deste algoritmo o próprio *router* pode ser agora DR ou BDR. A interface do *router* deve ser actualizada no estado DR. Se o router for BDR o estado deve ser actualizado como BDR. Se não for nenhum deles deve passar ao estado DROther.
6. Se a rede for NBMA e se o *router* tiver acabado de se tornar BDR ou DR deve começar a enviar mensagens Hello aos vizinhos que não forem elegíveis para se tornarem DR (prioridade igual a 0).
7. Se com os cálculos acima a identidade do DR ou BDR se alterarem o conjunto de adjacências associadas com a interface terão de ser modificadas.

# Eleição do *Designated Router* (DR)



- No segmento comum a RTA e RTB, como RTA e RTB têm a mesma prioridade e RTB tem um *router* ID maior, este último será o DR desse segmento.
- No segmento comum a RTB e RTC este último será o DR dado ter maior prioridade ( $P=2$ ).
- **Existirão tantos DR quantos os segmentos com dois ou mais *routers*.**



# OSPF – *Open Shortest Path First*



- Fundamentos
- Áreas
- Comunicação entre routers OSPF
- Vizinhos, Adjacências e *Designated Routers*
- Bases de dados de LSA: Sincronização e *Flooding*
- Rotas intra-área
- Rotas inter-área
- Rotas inter-AS
- Exemplos: rotas intra-área
- Exemplos: rotas inter-área

# Tipos de LSA (*link type*)

---



- **Type 1 – Router LSA:** O *router* descreve as redes a que está ligado e respectivos custos (interfaces)
- **Type 2 – Network LSA:** O DR descreve quais *routers* ligados a uma dada rede BMA ou NBMA
- **Type 3 – Summary LSA:** São enviados pelos ABR para descrever as redes de uma área
- **Type 4 – AS Summary LSA:** O ABR descreve a localização dos ASBR existentes nas área vizinhas
- **Type 5 – AS External LSA:** O ASBR descreve o sumário de destinos exteriores para os quais é Gateway

# Sincronização da BD de LSA

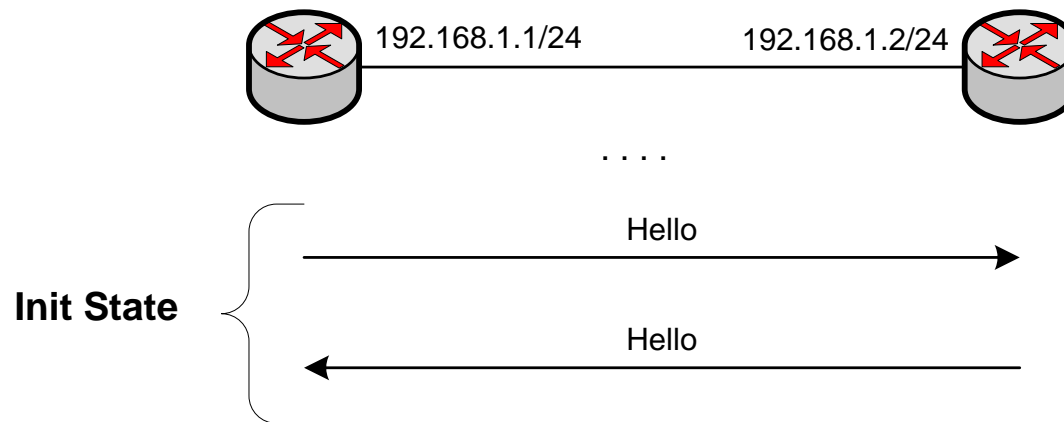


- **Estados** por onde passam dois *routers* para sincronização das suas BD de *link states*
  - **ExStart** - Envio de mensagens Database Description (DD) vazias para estabelecer a ligação (sincronizar o nº sequência) e escolha do Master
  - **Exchange** - Envio de mensagens Database Description (DD) indicando os LSA existentes na BD do *router*
    - Comparação entre as indicações recebidas e os LSA da BD local
  - **Loading** - Pedido dos LSA que necessitam de actualização através de Link State Requests
  - **Loading** - Actualização da BD com os LSA recebidos nos Link State Updates
  - **Full** - BD sincronizadas

# Sincronização da BD de LSA: *Hello*



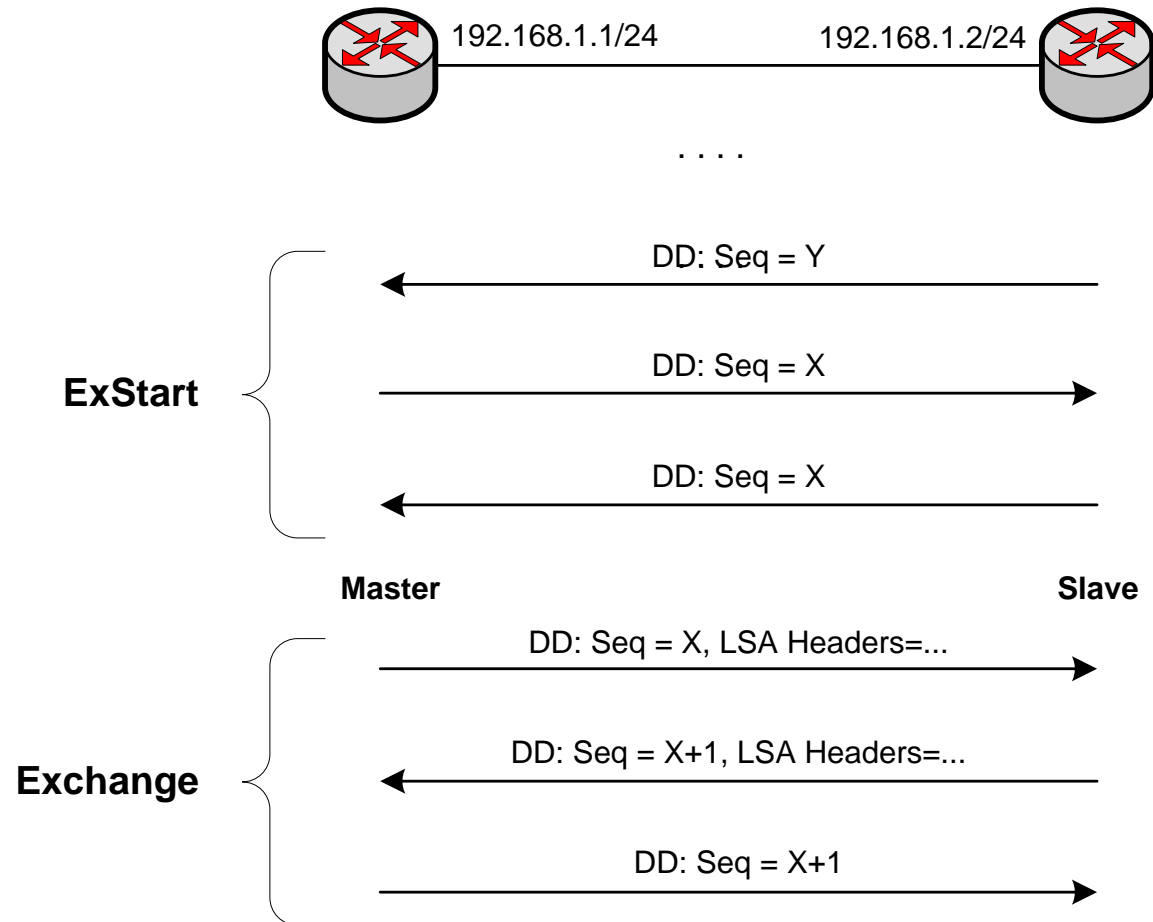
- Descoberta dos vizinhos; Escolha do *Designated Router* e criação de adjacências;



# Sincronização da BD de LSA: ExStart e Exchange



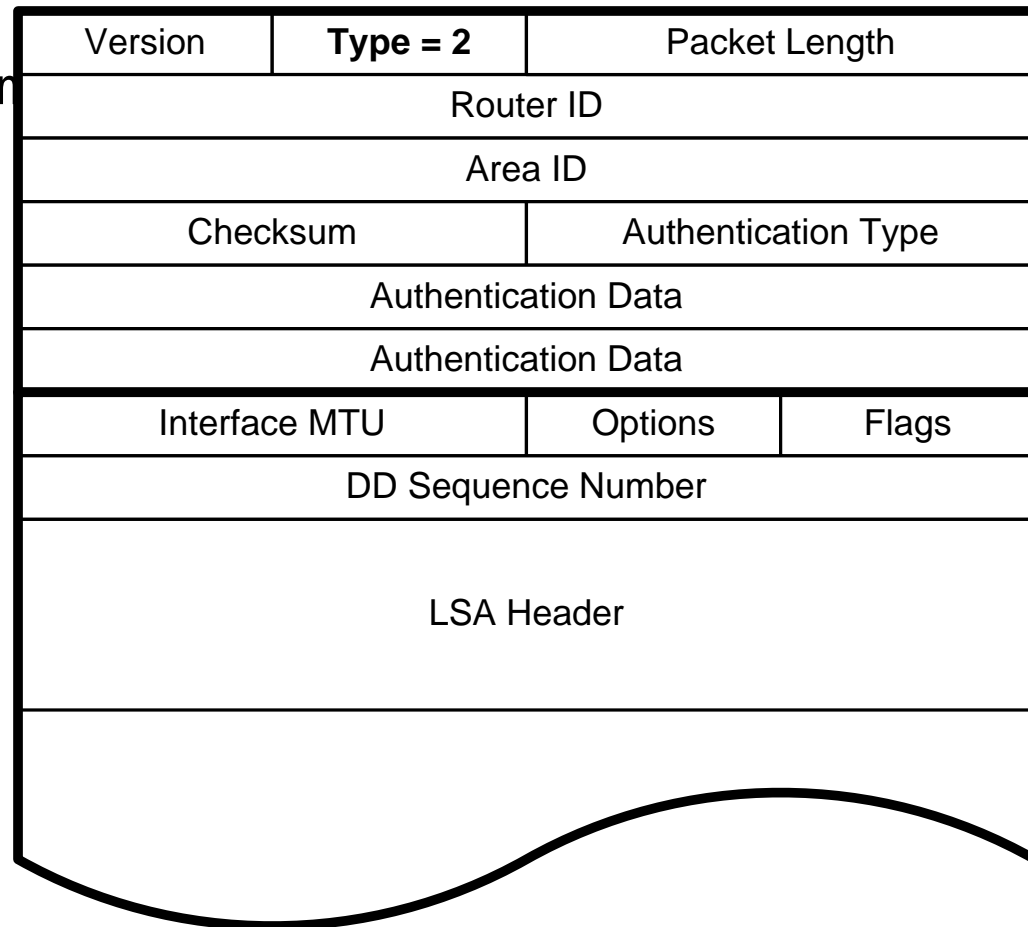
- *Router* direito torna-se Master porque possui um *router* ID maior. O *router* esquerdo torna-se Slave
- Os vizinhos trocam e confirmam pacotes de descrição da base de dados (DD).
- A verificação é efectuada através do número de sequência dos pacotes DD



# Mensagem *Database Description* (DD)



- Esta mensagem transporta apenas a descrição da base de dados, e não o seu conteúdo, de maneira a permitir que *routers* adjacentes possam comparar as suas bases de dados.
- Flag Initialize bit** = 1 nas primeira mensagem
- Flag More bit** = 0 na última mensagem
- Flag Master/Slave bit** - Indica quem começa e controla a sincronização dos LSA
- Ambos começam como Master e fica o que tiver o maior identificador
- DD Sequence Number** - Numera as mensagens DD enviadas por um *router*

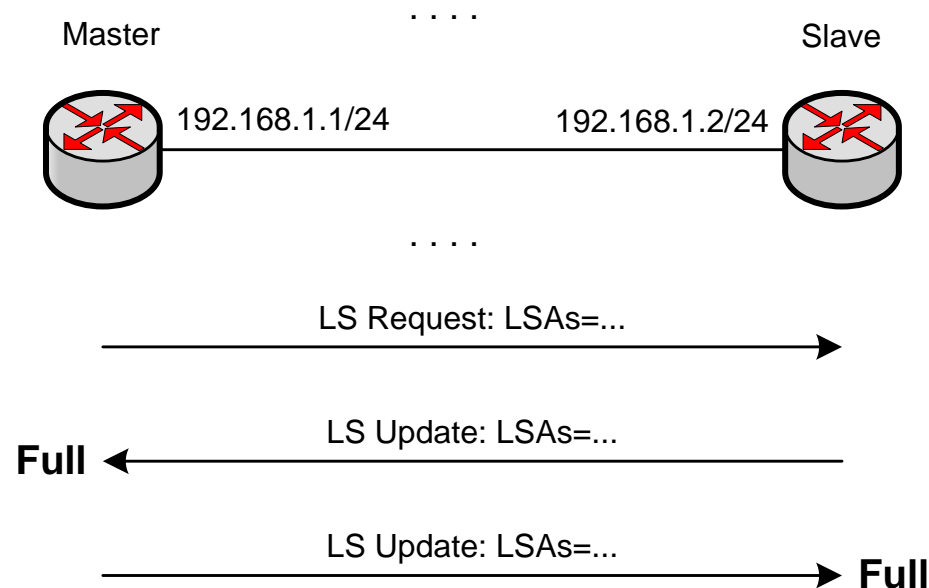




# Sincronização da BD de LSA: *Loading* e *Full*



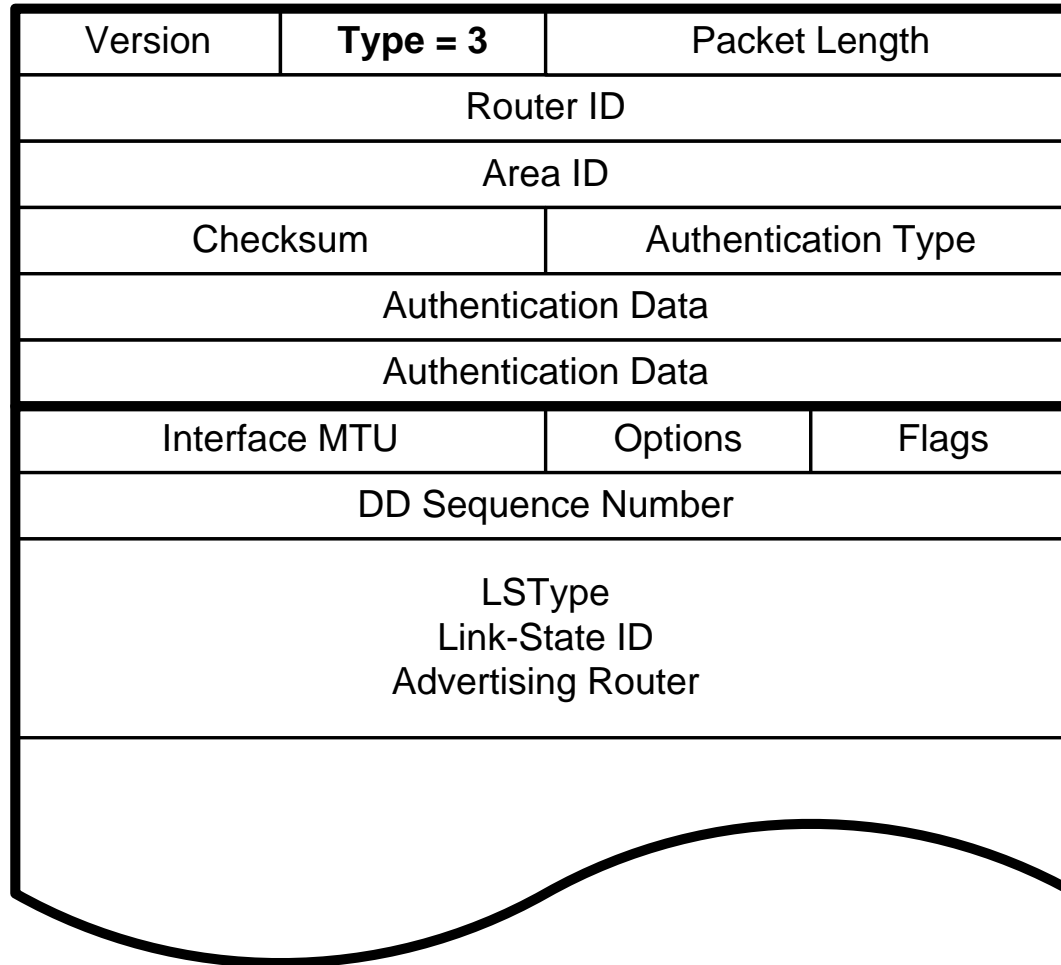
- Os pacotes de *Link State Request* são enviados para os vizinhos de forma a completar a informação na base de dados.



# Mensagem *Link State Request*



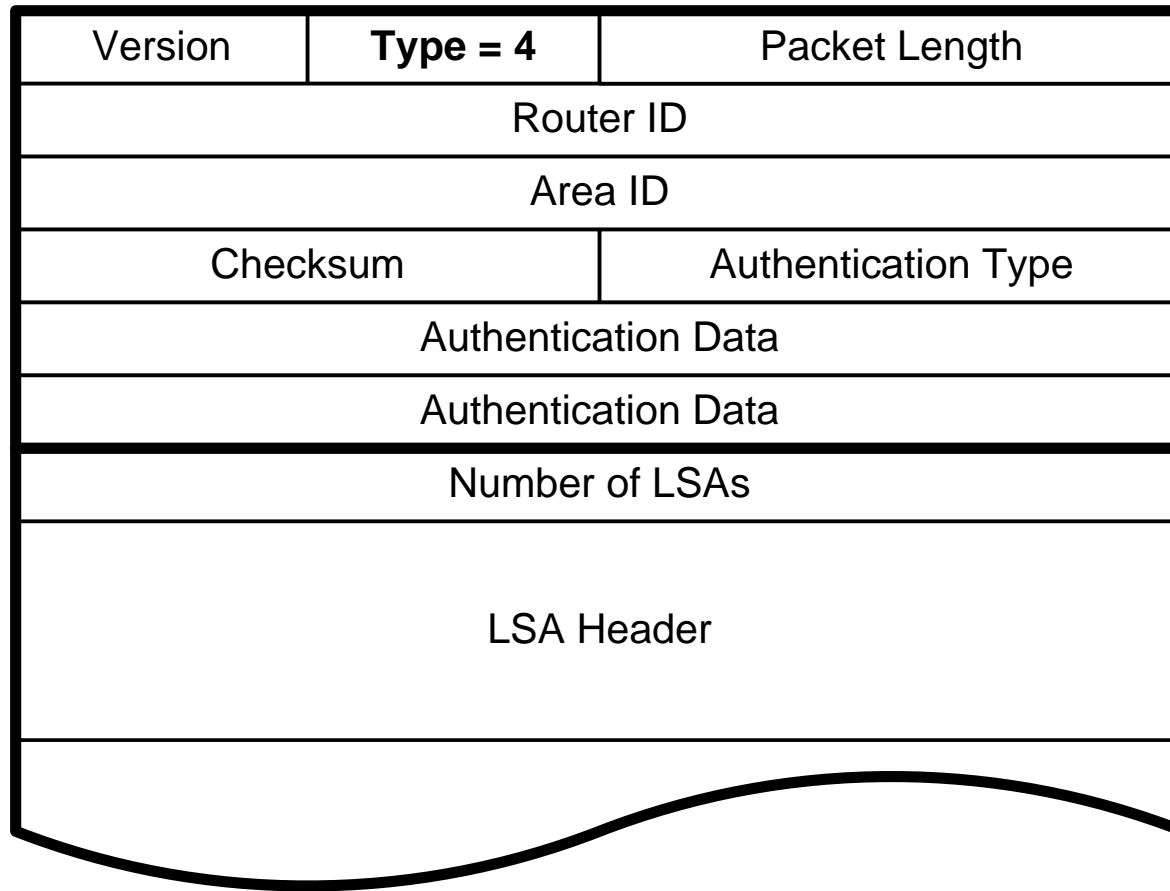
- Esta mensagem permite pedir a troca de LSA entre *routers* adjacentes.



# Mensagem *Link State Update*



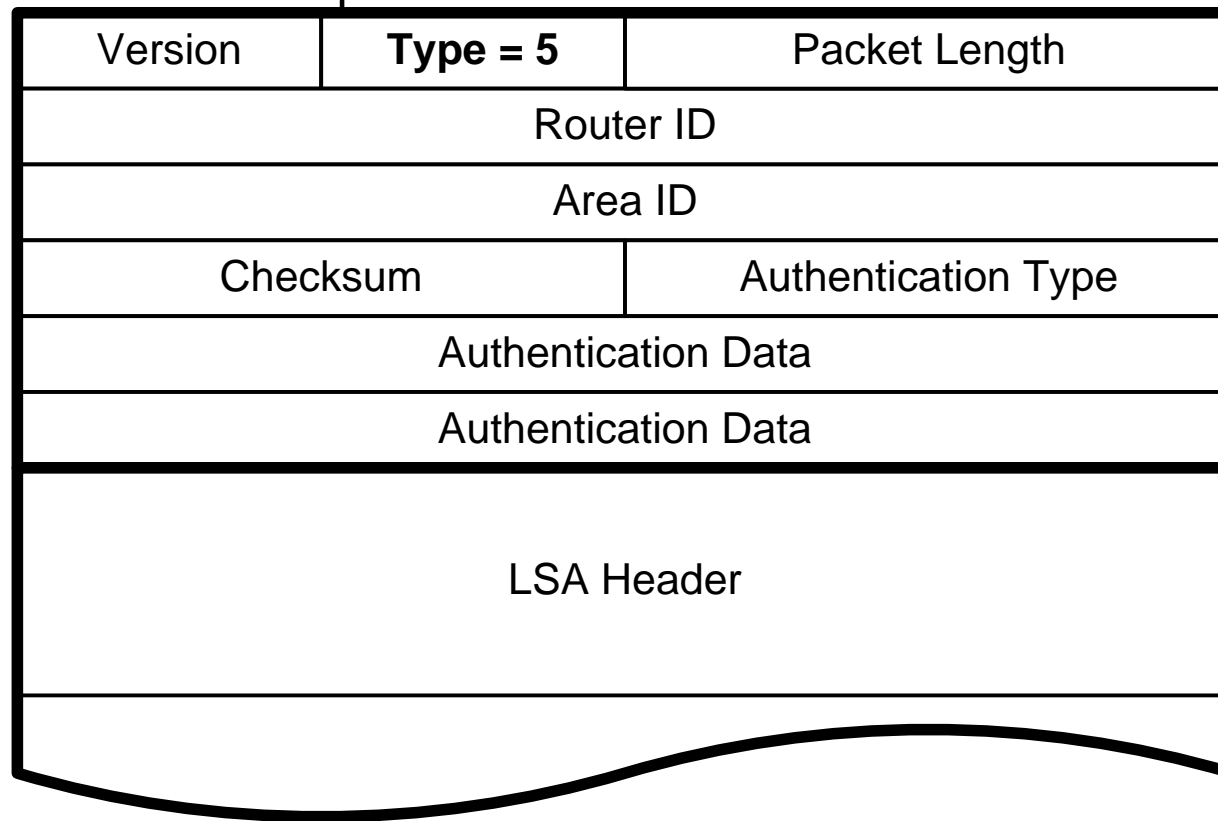
- Esta mensagem permite a troca de LSA entre *routers* adjacentes. Transporta um ou mais Link State Advertisement (LSA).



# Mensagem *Link Status Acknowledgement*



- Transporta *acknowledges* de um ou mais *Link State Advertisement* (LSA).
- Os LSA têm *acknowledge* individuais embora possam coexistir vários *acknowledges* num mesmo pacote de LSack.



# Estrutura dos LSA - *Header*



LS age	Options	LS Type
Link-State ID		
Advertising Router ID		
LS Sequence Number		
LS Checksum		Length

- **LS Age** - Tempo em segundos desde que o LSA foi gerado (actualizado nas BD)
- **Options** - T (suporta TOS)[obsoleta], E (external routing), P (NSSA ext.)
- **LS Type** - Router [1], Network [2], Summary [3 e 4], External [5], NSSA Ext. [7]
- **Link State ID** - Identificador do LSA (a interpretação deste campo varia conforma o *link type*)
- **Advertising Router** - Identificador do *router* que gerou o LSA
- **LS Sequence Number** - Versão do LSA, para detectar LSA antigos ou duplicados ( $-2^{31} + 1$  a  $2^{31} - 1$ )
- **LS checksum** – *Checksum* de todo o *header*, exceptuando o *Link Age*
- **Length** - Tamanho da mensagem em bytes

# Interpretação do campo “*link state ID*” dos LSA



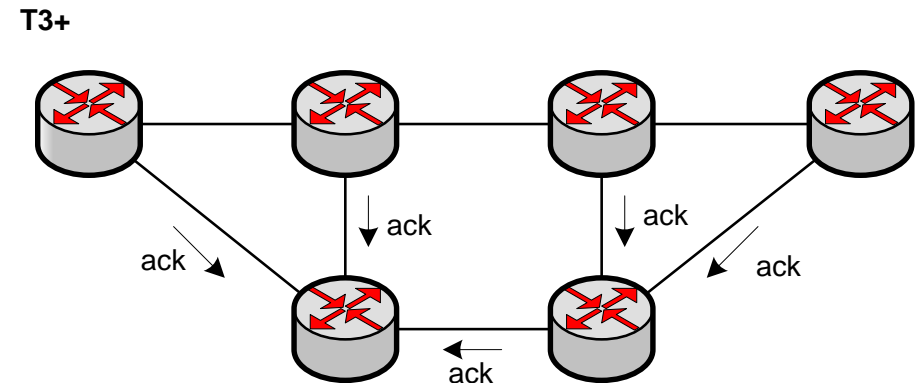
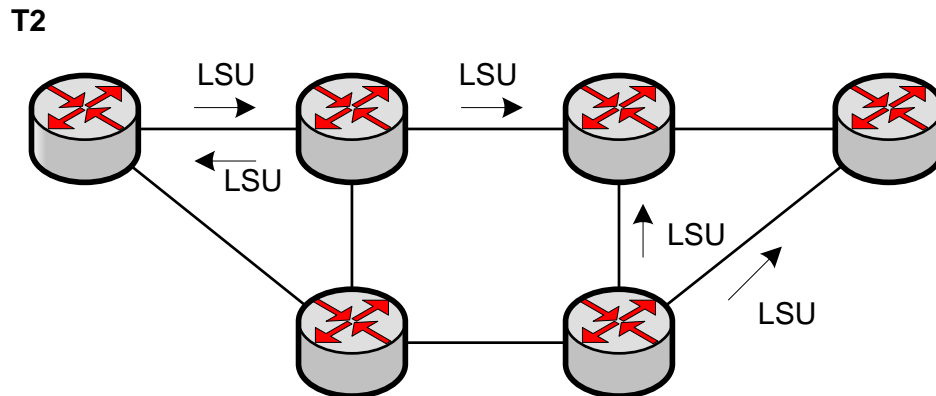
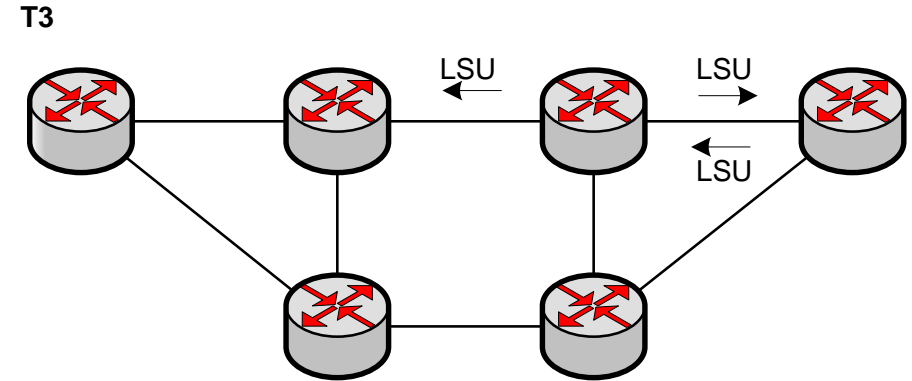
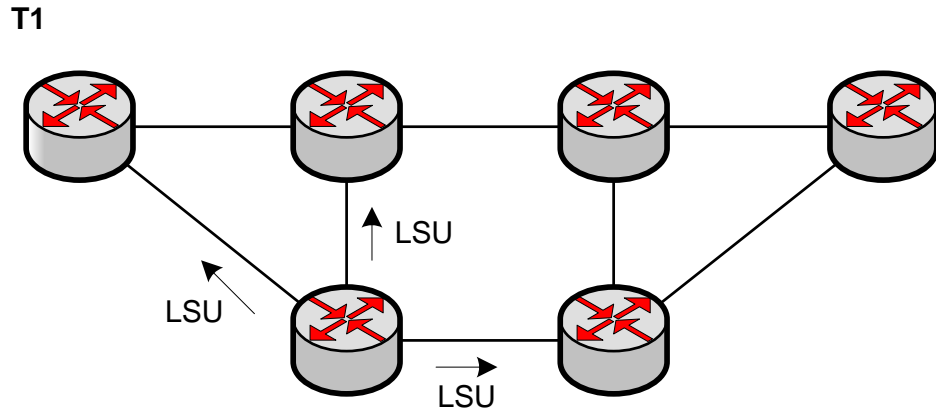
- Parte do domínio de *routing* que é descrito pelo LSA. Dependendo do *LS type* do LSA o “*Link State ID*” deve ser interpretado da seguinte forma:
  - *LS type* = 1: *Router ID* do *router* de origem.
  - *LS type* = 2: Endereço IP da interface do DR da rede (segmento).
  - *LS type* = 3: Endereço IP da rede de destino.
  - *LS type* = 4: *Router ID* do *ASBR* descrito.
  - *LS type* = 5: Endereço IP da rede de destino.

# Actualização da BD de LSA: *Flooding*



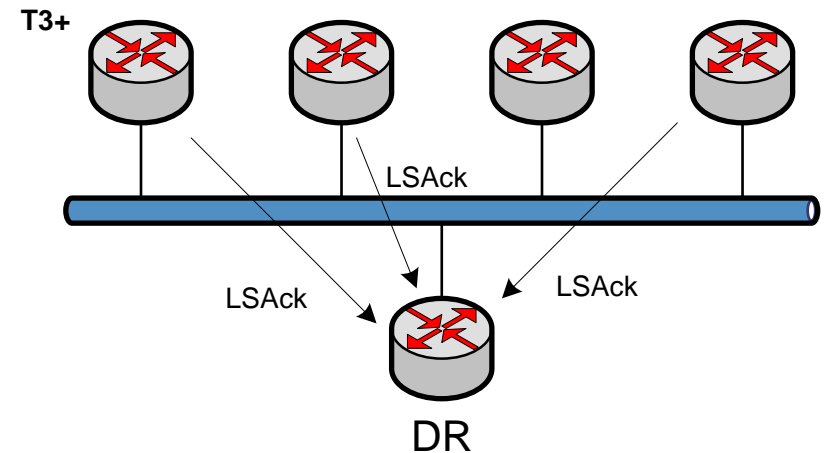
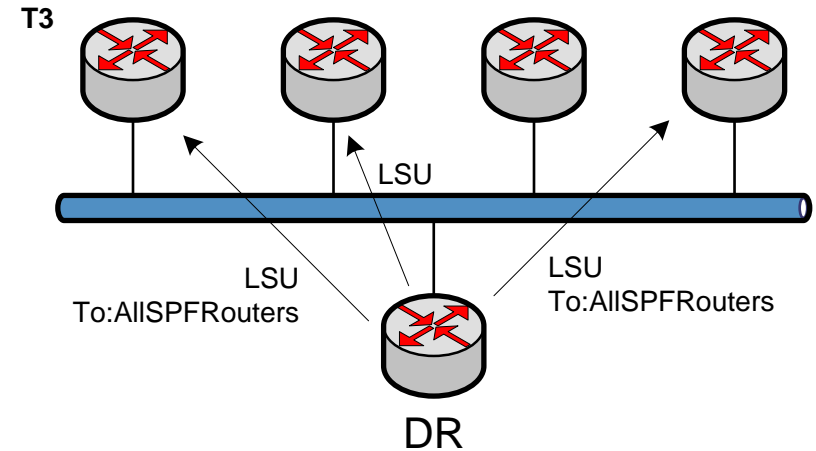
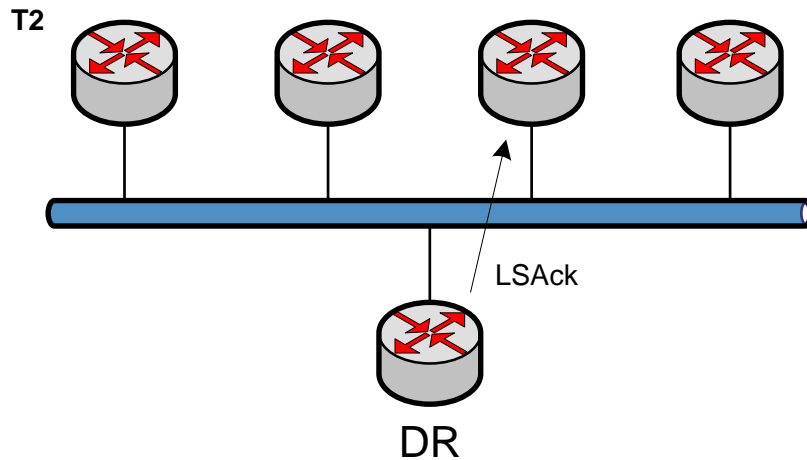
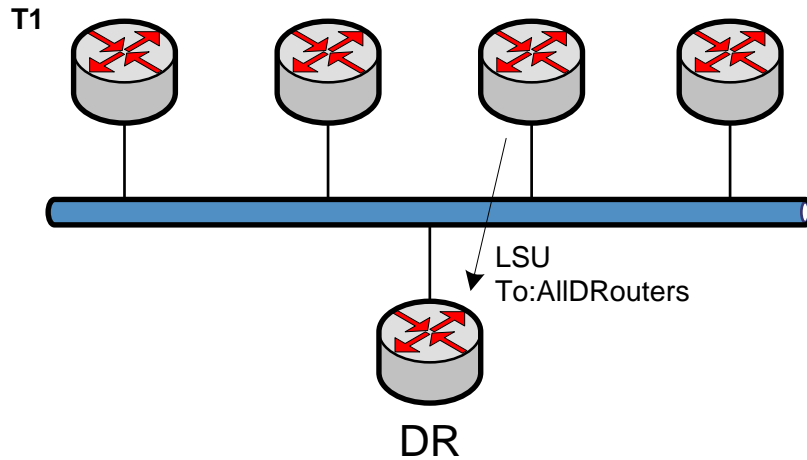
- Consiste em propagar os LSA por todas as adjacências de modo fiável
  - Envio de mensagens *Link State Update*
    - Campos *Sequence Number*
  - Mecanismo de *timeout* e retransmissão
  - Receptor actualiza a *link state database* e confirma a recepção através do envio de mensagens *Link State Acknowledge*
    - Campos *Sequence Number* e de *Link Age*
- As actualizações de LSA são enviadas apenas quando há uma alteração ou a cada 30 minutos
- Muito rápido e muito fiável, mas gasta largura de banda
- Os LSA são transportados em mensagens *Link State Update*.

# Flooding em redes ponto-a-ponto





# Flooding em redes *broadcast*





# Segurança das actualizações dos “mapas”

---

- O procedimento de *flooding* inclui as confirmações (*acknowledgments*) *hop-by-hop*
- As mensagens de descrição da base de dados são transmitidas de uma forma segura
- Cada registo (LSA) é protegido com um *timer* e é removido da base de dados se uma mensagem a refrescá-lo não aparecer no devido tempo
- Todos os registos são protegidos por um *checksum*
- As mensagens podem ser autenticadas, por exemplo através de *passwords*.